

VIII.3.3-RES-SNGL-PO4 SINGLE RESERVOIR REGULATION OPERATION USER
VARIABLE STORAGE FORMAT

This Section describes the format in which user variables are stored in the PO array for Operation RES-SNGL.

The starting location within the PO array of the start of the user variable information is held in word 13 of the PO array. A value of zero indicates that no user variables are stored.

User variables are referred to by location from an IF EXPRESSION (see Section VIII.3.3-RES-SNGL-PO3 for additional details).

The first word in this section of the PO array holds the number of user variables that are stored. The remainder is filled with the necessary information to describe the user variables.

There are three types of user variables:

- o Type 1 - a constant value with no name
- o Type 2 - a constant value with a name
- o Type 3 - a system function

The constant value without a name, Type 1, is one where the number used for comparison is entered directly in the IF phrase (during PIN input, that is). Type 2, the constant value with a name, is defined through the use of the RCL SET statement (see Section V.3.3-RES-SNGL). The system function, Type 3, is either RULE or MAXQ.

The information stored for Type 1 user variables is the type code (=1.01), the value itself and the unit conversion code for a total of three words. This last piece of information is necessary for converting the value from input units to metric units and for regenerating the IF expressions whenever the operation definition is to be displayed. This code is also used for Type 2 user variables. The codes stored for the system variables used as the left-hand-side of the comparisons are:

<u>System Variable</u>	<u>Unit Dimension</u>	<u>Unit Conversion Code</u>
QO	L3/T	1.01
QI	L3/T	1.01
POOL	L	2.01
STORAGE	L3	3.01
DAY	N/A	0.01
QOM	L3/T	1.01
QIM	L3/T	1.01

For Type 2 user variables, six words of information are stored: the type code (=2.01), the 12 character name (three words padded with any necessary blanks), the value itself and the unit conversion code (as explained under Type 1 discussion).

The Type 3 user variable has stored for it four words of information: the type code (=3.01), the numerical code of the system function, a

code for any mathematical operator working on the function and the mathematical factor used with the operator. Two functions are available:

System <u>Function</u>	Numerical <u>Code</u>
RULE	610.01
MAXQ	1570.01

The code actually stored for the MAXQ function is the code listed plus its level of definition as MAXQ is a utility that can have multiple definitions. (see Section VIII.3.3-RES-SNGL-PO1 for additional information on Scheme/Utility base and level numbers).

The mathematical operators and codes are:

Mathematical <u>Operator</u>	Numerical <u>Code</u>
None	0.01
+	1.01
-	2.01
*	3.01
/	4.01

The mathematical factor is stored and used only if the operator is nonzero and is just the value properly converted from input units to internal storage units (in units of M for RULE and CMS for MAXQ). The conversions are determined to be the function being operated upon.

User variables for the examples given above are stored as follows:

1. QO.GT.50000.00

This is a Type 1 user variable. Three values are stored:

1.01	50000.00	1.01
Type	Value	Unit Conversion Code for flow

2. QOM.LT.MAXMEAN

This is a Type 2 user variable. Six words are stored:

2.01	'MAXMEAN'	value	1.01
Type	Name (3 word)		Unit Conversion Code for

flow

3. DAY.GT.31

This is a Type 1 dimensionless user variable. Three values are stored:

1.01	31	0.01
Type	Value	Unit Conversion Code

4. POOL.GE.RULE+2.0 (2 feet = 0.61 meters)

This is a Type 3 user variable. Four values are stored:

3.01	610.01	1.01	0.61
Type	Function	Mathematical	Factor in Meters
	Code of Rule	Operator '+'	

User variables are referred to by location from an IF EXPRESSION as explained by the following example:

```
RCL
  IF (expression 1) THEN DO POOLQ
  ELSEIF (expression 2) THEN DO POOLQ
  "      (      "      3)      "      "
  "      (      "      4) DO SETQ(2)
  DO SETQ(3)
  DO SETMIN
  ELSEIF (expression 5) THEN DO SETH
  ELSE DO POWERGEN
  DO RULECURVE
  DO SETQ
  ENDIF
END RCL
```

```
(expression 1) = (DAY.GE.1.00.AND.DAY.LT.60.00.AND.
                 POOL.GT.RULE+2.00.AND.POOL.GT.480.00
                 .AND.POOL.LT.490.00)
(expression 2) = (DAY.GE.60.00 .AND.DAY.LT.90.00.AND.
                 POOL.GT.RULE+3.00.AND.POOL.LT.490.00)
(expression 3) = (DAY.GTE.90.00.AND.DAY.LE.366.00.AND.
                 POOL.GT.RULE+2.00.AND.POOL.LT.490.00)
(expression 4) = (POOL.GE.490.00.AND.POOL.LT.490.50)
(expression 5) = (POOL.GE.490.50)
```

The PO array encoded form of eight IF EXPRESSION groups in the IF EXPRESSION section of PO array is:

<u>8.01</u>	<u>19.01</u>	<u>505.01</u>	<u>-2.01</u>	<u>166.01</u>	<u>505.01</u>	<u>-5.01</u>	<u>167.01</u>
0	<u>1/</u>	DAY	U.V.POINTER	GE	DAY	U.V.POINTER	LT
<u>149.01</u>	<u>503.01</u>	<u>-8.01</u>	<u>164.01</u>	<u>149.01</u>	<u>503.01</u>	<u>-12.01</u>	<u>164.01</u>
AND	POOL	U.V.POINTER	GT	AND	POOL	U.V.POINTER	GT
<u>149.01</u>	<u>503.01</u>	<u>-15.01</u>	<u>167.01</u>	<u>149.01</u>	<u>15.01</u>	<u>505.01</u>	<u>-18.01</u>
AND	POOL	U.V.POINTER	LT	AND	<u>2/</u>	DAY	U.V.POINTER
<u>166.01</u>	<u>505.01</u>	<u>-21.01</u>	<u>167.01</u>	<u>149.01</u>	<u>503.01</u>	<u>-24.01</u>	
GE	DAY	U.V.POINTER	LT	AND	POOL	U.V.POINTER	
<u>164.01</u>	<u>149.01</u>	<u>503.01</u>	<u>-28.01</u>	<u>167.01</u>	<u>149.01</u>	<u>15.01</u>	<u>505.01</u>
GT	AND	POOL	U.V.POINTER	LT	AND	<u>3/</u>	DAY
<u>-31.01</u>	<u>166.01</u>	<u>505.01</u>	<u>-34.01</u>	<u>165.01</u>	<u>149.01</u>	<u>503.01</u>	
U.V.POINTER	GE	DAY	U.V.POINTER	LE	AND	POOL	
<u>-37.01</u>	<u>164.01</u>	<u>149.01</u>	<u>503.01</u>	<u>-41.01</u>	<u>167.01</u>	<u>149.01</u>	
U.V.POINTER	GT	AND	POOL	U.V.POINTER	LT	AND	

<u>7.01</u>	<u>503.01</u>	<u>-44.01</u>	<u>166.01</u>	<u>503.01</u>	<u>-47.01</u>	<u>167.01</u>	<u>149.01</u>
<u>4/</u>	POOL	U.V.POINTER	GE	POOL	U.V.POINTER	LT	AND
<u>3.01</u>	<u>503.01</u>	<u>-50.01</u>	<u>166.01</u>				
<u>5/</u>	POOL	U.V.POINTER	GE				

Notes:

PO(PO(14)); IF EXPRESSION groups begins;
8 IF EXPRESSION groups are used

- 1/ PO(PO(14)+1); (expression 1) begins; 19 values used
- 2/ PO(PO(14)+19+1); (expression 2) begins; 15 values used
- 3/ PO(PO(14)+19+15+1); (expression 3) begins; 15 values used
- 4/ PO(PO(14)+34+15+1); (expression 4) begins; 7 values used
- 5/ PO(PO(14)+49+7+1); (expression 5) begins; 3 values used

The PO array encoded form of 29 user variables in the USER VARIABLE section of PO array is:

<u>1/16.01</u>	<u>1/1.01</u>	<u>1.00</u>	<u>0.01</u>	<u>2/1.01</u>	<u>60.01</u>	<u>0.01</u>	
1	2	3	4	5	6	7	
<u>3/3.01</u>	<u>610.01</u>	<u>1.01</u>	<u>0.61</u>	<u>4/1.01</u>	<u>146.30</u>	<u>2.01</u>	
8	9	10	11	12	13	14	
<u>5/1.01</u>	<u>149.35</u>	<u>2.01</u>		<u>6/1.01</u>	<u>60.00</u>	<u>0.01</u>	
15	16	17		18	19	20	
<u>7/1.01</u>	<u>90.00</u>	<u>0.01</u>		<u>8/3.01</u>	<u>0.0</u>	<u>1.01</u>	<u>0.91</u>
21	22	23		24	25	26	27
<u>9/1.01</u>	<u>149.35</u>	<u>2.01</u>		<u>10/1.01</u>	<u>90.00</u>	<u>0.01</u>	
28	29	30		31	32	33	
<u>11/1.01</u>	<u>366.00</u>	<u>0.01</u>		<u>12/3.01</u>	<u>610.01</u>	<u>1.01</u>	<u>0.61</u>
34	35	36		37	38	39	40
<u>13/1.01</u>	<u>149.35</u>	<u>2.01</u>		<u>14/1.01</u>	<u>149.35</u>	<u>2.01</u>	
41	42	43		44	45	46	
<u>15/1.01</u>	<u>149.50</u>	<u>2.01</u>		<u>16/1.01</u>	<u>149.50</u>	<u>2.01</u>	
47	48	49		50	51	52	

Notes:

PO(PO(13)); USER VARIABLE group begins;
16 USER VARIABLE groups are used

- 1/ User variable group 1 begins at position 2;
Type = 1, value = 1 day
- 2/ User variable group 2 begins at position 5;
Type = 1 then value = 60 day

3/ User variable group 3 begins at position 8;
Type = 3, system function = RULE, factor = 0.61 M (2.00 FT)

4/ User variable group 4 begins at position 12;
Type = 1, value = 146.30 M (480 FT)

5/-16/ For user variable groups 5-16.